**Exercises JSON**

Download the start files before starting with these exercises.

**Exercise 1**

The file *PythonCoursesTM.json* contains a list of courses at Thomas More in which Python is used as programming language. Besides the course itself, the class groups who follow these courses are also included.

Write a program that will read out the courses and their class groups and display them as follows:

Graphical user interface, text

Description automatically generated

Note that class groups are separated by a tab and that the courses are separated by a blank line.

**Exercise 2**

Make a new program in which you use the *PythonCoursesTM.json* file again. This time, read out all the groups and the number of students of each group and display them as shown below. Make sure groups are only shown once (groups might follow more than one course). You may assume that groups have the same amount of students in each course they follow.

Sort the result alphabetically, based on the group’s name.

Afbeelding met tekst, persoon, document

Automatisch gegenereerde beschrijving

**Exercise 3**

The school wants to make an inventory of the available space in each room. Write a program that will allow an employee of the facilities department to enter a room number, followed by the theoretical capacity of that room. Your program should also ask for the following properties of each room:

* Blackboard available
* Beamer available
* Desktop available

The user will answer to the questions above with Y (yes) or N (no). Your program may not be case sensitive. If the user enters another value, your program should assume the answer is N (no).

After entering all the required information, your program should allow the user to enter remarks concerning that room.

The facilities employee can keep entering room information until the word ‘end’ is entered. Your program should recognize this word both in lower and upper case letters.

Pay attention to the details you can see in the screenshot below.

Text

Description automatically generated

Once the user input stops, your program should generate the file *rooms.json* which contains all necessary information. Pay attention to the data types when creating this JSON file, the JSON structure and data types must match with the structure shown below.

Text

Description automatically generated

**Exercise 4**

The file *cocktail.json* contains recipes for some cocktails. Create a program which will read out this file and ask the user for which cocktail (s)he would like to see the recipe. Show the available cocktails in the following way:

Text, letter

Description automatically generated

Next, ask the user which cocktail should be shown. You may assume that the user will always enter a number from the list. After entering the user’s choice, show the chosen cocktail, its ingredients (including measures) and its recipe (instructions) as shown below.

Text

Description automatically generated with low confidence

